Integrating Manual and Automatic Annotation for the Creation of Discourse Network Data Sets

Haunss, Sebastian; Kuhn, Jonas; Padó, Sebastian; Blessing, Andre; Blokker, Nico; Dayanik, Erenay; Lapesa, Gabriella

Veröffentlichungsversion / Published Version
Zeitschriftenartikel / journal article

COGITATIO

Politics and Governance (ISSN: 2183–2463)
2020, Volume 8, Issue 2, Pages 326–339
DOI: 10.17645/pag.v8i2.2591

Article

Integrating Manual and Automatic Annotation for the Creation of Discourse Network Data Sets

Sebastian Haunss [1,*], Jonas Kuhn [2], Sebastian Padó [2], Andre Blessing [2], Nico Blokker [1], Erenay Dayanik [2] and Gabriella Lapesa [2]

[1] Research Center on Inequality and Social Policy, University of Bremen, 28359 Bremen, Germany;
E-Mails: sebastian.haunss@uni-bremen.de (S.H.), blokker@uni-bremen.de (N.B.)
[2] Institute for Natural Language Processing, University of Stuttgart, 70569 Stuttgart, Germany;
E-Mails: jonas.kuhn@ims.uni-stuttgart.de (J.K.), sebastian.pado@ims.uni-stuttgart.de (S.P.), andre.blessing@ims.uni-stuttgart.de (A.B.), erenay.dayanik@ims.uni-stuttgart.de (E.D.), gabriella.lapesa@ims.uni-stuttgart.de (G.L.)

* Corresponding author

Submitted: 30 October 2019 | Accepted: 23 March 2020 | Published: 2 June 2020

Abstract
This article investigates the integration of machine learning in the political claim annotation workflow with the goal to partially automate the annotation and analysis of large text corpora. It introduces the MARDY annotation environment and presents results from an experiment in which the annotation quality of annotators with and without machine learning based annotation support is compared. The design and setting aim to measure and evaluate: a) annotation speed; b) annotation quality; and c) applicability to the use case of discourse network generation. While the results indicate only slight increases in terms of annotation speed, the authors find a moderate boost in annotation quality. Additionally, with the help of manual annotation of the actors and filtering out of the false positives, the machine learning based annotation suggestions allow the authors to fully recover the core network of the discourse as extracted from the articles annotated during the experiment. This is due to the redundancy which is naturally present in the annotated texts. Thus, assuming a research focus not on the complete network but the network core, an AI-based annotation can provide reliable information about discourse networks with much less human intervention than compared to the traditional manual approach.

Keywords
annotation; automation; discourse networks; machine learning; migration discourse

Issue
This article is part of the issue “Policy Debates and Discourse Network Analysis” edited by Philip Leifeld (University of Essex, UK).

1. Introduction

Discourse network analysis (DNA) offers a conceptual framework for the analysis of discourse structures and dynamics. Numerous DNA studies have shown that the network perspective on political discourse offers insights that go beyond traditional policy analyses and qualitative discourse studies (Haunss, 2017; Leifeld, 2016; Nagel & Satoh, 2019; Wang & Wang, 2017). In principle, modelling the development of political debates as dynamic discourse networks may enable us to identify recurring mechanisms that drive the development of political debates and to distinguish between network effects and actor attribute effects. Unfortunately, the creation of dynamic discourse network data sets is extremely time- and labour-intensive and therefore poses a serious barrier for this kind of analysis.

In this article, we present the first results from a research project in which we investigate annotation workflows that integrate machine learning to partially automate and thus significantly speed up the annotation of large text corpora. The article addresses two closely related research questions: First, it asks to what extent the integration of machine learning tools can enhance annotation by human annotators in terms of annotation speed and annotation quality; second, it evaluates the quality of the discourse network representation of the machine learning based annotations. This allows us to fully assess the potential of our (semi-)automatic methodology.

Regarding the first question, we present results from an annotation experiment that, indeed, show overall gains in terms of annotation speed, and a moderate increase in annotation quality with the assistance of machine learning based predictions. Additionally, given the increase in annotation quality, the approach might help to reduce bias in the generation and analysis of discourse networks by increasing the number of claims found, which otherwise would not have been identified by the human annotators.

Regarding the second question, we compare the discourse networks that would result from the annotations of a machine learning based automatic pseudo-annotator, and where human coders would only eliminate false positives, with those discourse networks resulting from our manual annotation. In this setting, our system performs surprisingly well, and we can show that it is possible to reproduce the core discourse network with only minimal manual intervention. While these findings are still preliminary and abstract from still open tasks of reliable automatic speaker identification and fine-grained claim classification, they open up new opportunities for semi-automatic annotations of large text corpora.

We first present our modelling approach and discuss our strategy to integrate machine learning for claim identification and claim categorisation. In the second part of the article, we report results from an experiment in which the annotation quality of annotators with and without machine learning based annotation support is compared. Finally, we discuss the potential for a more automated annotation model by evaluating the experimental data with discourse networks.

2. Existing Approaches to Analyse the Content of Large Text Corpora in the Social Sciences

In the social sciences and humanities, analysis of text corpora typically distinguishes between qualitative and quantitative approaches, or a mixture of both (Kelle, 2008; Kuckartz, 2014). However, when dealing with large text corpora, text analysis is always quantitative because it bases its argumentation necessarily on some form of numeric evaluation of the text data. The main difference between the various approaches is whether they rely mainly on statistical evaluation of the raw textual data or whether they include some form of content-based abstraction from the original text.

The first group of these approaches comprises text mining (TM) techniques that rely on word frequency, co-occurrence analysis, or on the analysis of the distribution of syntactic patterns at the text surface (which serve as an indication for underlying information, e.g., social group membership). From this perspective, texts are viewed as sets of such surface cues, and TM tries to directly draw conclusions from the statistical distribution of these cues (Wiedemann, 2016, p. 40). This opens the possibility to quickly analyse large corpora, which cannot be researched manually in a reasonable timeframe. Studies in this vein have been able to automatically identify actors' policy positions on a political left–right scale (Laver, Benoit, & Garry, 2003) or support vs. opposition to legislative proposals (Klüver, 2009). They can identify topics in political debates and explore the structure in which these topics are related (Walter & Ophir, 2019), and analyse the tone of political debates using sentiment analysis (Burscher, Vliegenthart, & de Vreese, 2016). Recent work combines machine learning with more traditional statistical approaches (for an overview see Welbers, van Atteveldt, & Benoit, 2017; for a discussion see Wilkerson & Casas, 2017).

The second group of approaches tries to capture complex meaning structures on a more fine-grained level. They usually rely on more or less extensive annotation of the raw text material by human annotators, following a codebook that provides categories at a certain level of abstraction from the original text in order to identify political claims (Koopmans & Statham, 2010), frames (D'Angelo & Kuypers, 2010), or evaluative statements (Schmidtke & Nullmeier, 2011). Although manual text annotation offers very precise results, it is extremely expensive. Quantitative annotation-based text analysis therefore usually tries to scale up a reduced set of techniques from qualitative text analysis, notably the assignment of abstract categories to text segments.

Various combinations of TM and annotation approaches have been suggested, where TM is used to structure the corpus and to answer more general research questions, and where only a limited sub-set of texts is then manually annotated, effectively reducing the amount of annotated text (Stulpe & Lemke, 2016). The methodological approach we present in this article follows a different logic. It places considerable emphasis on careful manual annotation (and codebook development) but takes advantage of recent machine learning techniques. Only a comparatively small set of text data is initially manually annotated without machine learning support, and this is then used as training data for classifiers that can expand the scope of analysis to considerably larger corpora. Instead of limiting the amount of annotated text, we aim at annotating the complete corpus but limiting the amount of manual annotation without machine learning support. The limited precision and recall of machine-learned classifiers can be counteracted in a 'mixed methods' approach: Where precision is important, automatic predictions are not used to replace manual annotation decisions, but to speed up the process. Where the corpus includes enough redundancy, aggregation over automatic predictions can make up for recall issues.

3. MARDY: The Task, the Challenges and the Annotation Environment

The MARDY (Modeling ARgumentation DYnamics in political discourse) annotation environment enables parallel multi-user annotation of texts and the integration of machine learning based annotation (the software components of the MARDY environment are listed in Appendix 1 in the Supplementary File; for a detailed description see Blessing et al., 2019). In the specific study presented here, we use it to annotate political claims in newspaper articles in the German daily quality newspaper *taz—die tageszeitung*. Drawing on Koopmans and Statham (2010, p. 55), we define a claim as a purposeful communicative action in the public sphere by which an actor tries to influence a specific policy or political debate. A claim can be a verbal statement or another form of action like a protest or a political decision that articulates political demands, calls to action, proposals, or criticisms.

Manual claim annotation involves multiple steps. Claims need to be identified in the text, a speaker/actor needs to be identified and assigned to the claim, and the identified claim needs to be assigned a category, a polarity (support or opposition) and a date (by default the day before the publication of the article). With MARDY we ask two questions: (a) What would this process look like if we could automate it completely?; and (b) how can we digitally support manual annotation?

The answer to (a) is shown in the left-hand panel of Figure 1. The annotation steps can be mapped fairly directly to tasks that a completely automatic discourse network extraction system would have to carry out. Arguably, an automatic system should not have to predict the date; meanwhile, it makes sense to include the aggregation step (moving from individual annotations to a network) into its purview.

With regard to (b), a computer-supported annotation environment can help the annotation process on four levels: 1) speed up the manual annotation process; 2) support the conceptual side of the annotation process; 3) improve annotation quality and consistency; and 4) (partially) automate the annotation process by integrating machine learning for claim detection and classification. We will now give short sketches of the first three points and then discuss how the MARDY annotation environment integrates machine learning in more detail (links to a demo version of the annotation environment and to the documentation and code are listed at the end of this article in Section 7).

The MARDY environment has the following goals:

Goal 1 (*speeding up the annotation*): To prevent the annotators from reading large amounts of irrelevant texts, MARDY performs document selection as a preprocessing step: By integrating a keyword and a document classification approach, MARDY shows to the annotators only documents that discuss the topic relevant for the annotation (i.e., in this article, immigration) and are therefore likely to contain claims. Thus, preprocessing speeds up the claim detection task effectively. Actor detection is also supported with pre-processing, as textual strings denoting potential actors are identified by employing automatic tools for named-entity recognition, stored in an updatable knowledge base, which was initialised by data records from Wikidata (Vrandečić & Krötzsch, 2014) and suggested to the annotator in the user interface.

Goal 2 (*conceptual annotation support*): In the lifecycle of an annotation project, annotators learn from the feedback of experts, and experts need to modify the initial classification scheme (the codebook) based on feedback from the annotators. MARDY supports both sides of

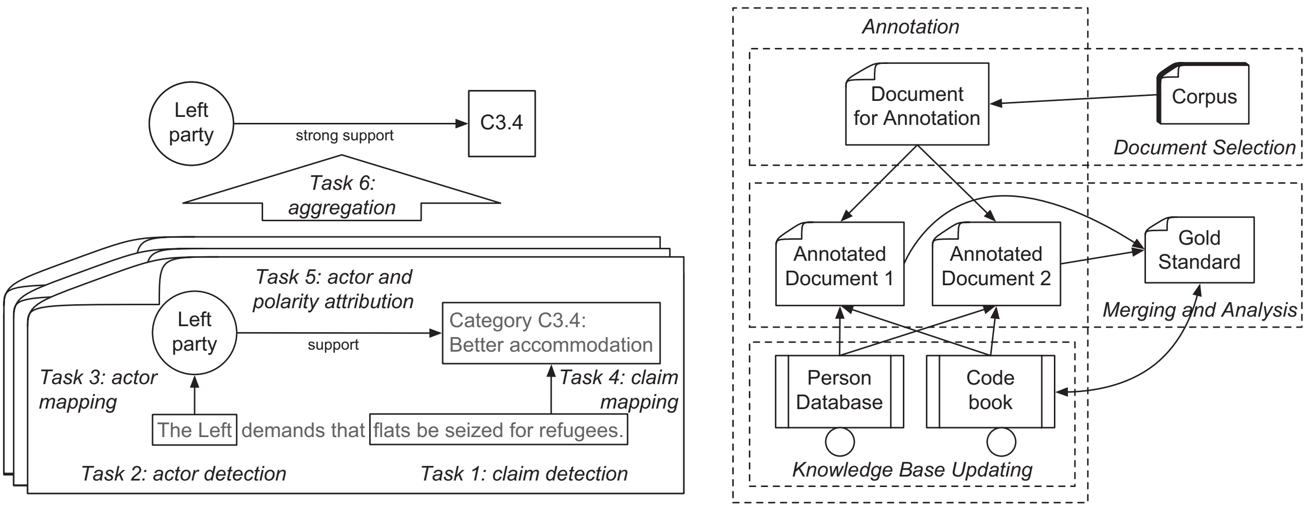

Figure 1. The MARDY approach to automatic discourse network creation. Notes: Processing steps in automatic prediction are in the left panel (adapted from Padó et al., 2019) and data flow for a manual annotation tool in the right panel (from Blessing et al., 2019).

this conceptual progress, as annotation performance can be constantly monitored. Individual performances and evaluations are available to both experts and (optionally) annotators in separate views which provide crucial statistics regarding annotation accuracy. Annotators can check their progress and, once the experts have revised their annotations, directly inspect the cases in which their annotation was suboptimal; experts can aggregate annotator errors by categories, thus uncovering trends and patterns which may suggest that the codebook needs to be updated to avoid systematic inconsistencies or points of confusion.

Goal 3 (*improving annotation quality in terms of consistency and coverage*): MARDY enables simultaneous annotation of the same document by multiple annotators via a browser-based user interface. The administration interface enables the experts to edit and merge the annotation performed by the annotators, leading to the creation of a reliable gold standard. In this stage, the expert acts as a super-annotator, who has the power of deleting/adding claims or adjusting span, category, actor, and polarity of the already detected claims.

Goal 4 (*integrating machine learning for claim detection and classification*): One incisive innovation compared to existing annotation frameworks in political science, such as MAXQDA, NVivo, Atlas.ti (Rädiker & Kuckartz, 2019), or DNA (Leifeld, 2009) is the integration of predictions of a machine learning classifier, which MARDY treats as a 'pseudo-annotator.' The pseudo-annotator takes over the tasks of claim detection and classification. Figure 2 displays the gold merging interface and it illustrates how it can be used by the experts to create the gold standard. On the left panel (blue rectangle), the system displays five candidate claims to be reviewed by the expert annotator. Candidate claims are of two types: They have been identified by a human annotator (annotator 20, marked with [20]: candidates 2, 4, and 5) or by the AI pseudo-annotator (marked with [AI]: claims 1 and 3). The panel in the middle shows the claims which were accepted into the gold standard. The panel on the right shows the text of the article; the expert has the possibility to highlight a specific candidate claim (in this example, candidate 1), thus retrieving a larger context without going through the entire article (function 'show' in the left panel). A pop-up window (activated with the 'edit' button in the central panel) allows to edit or change the details of the annotation: In this example, given that the [AI] classifies candidates into high-level categories (in this case, 800, 'Procedures'), the expert can introduce the finer-grained annotation (805, 'Additional Financing') as well as the actors and polarity. What we see in Figure 2 is a typical scenario, in which AI and human annotators turn out to be complementary. The first claim (corresponding to candidate 1) has been identified by AI and overlooked by the human annotator. The second and third claim has been identified by both human and AI, but with a different segmentation (one span for the AI, candidate 3; two spans for the human,

Figure 2. Merging interface for gold standard, including AI suggestions.

candidates 2 and 4). The fourth claim has been identified only by the human annotator (candidate 5).

At this point, a natural question to ask is how good the AI annotator is. We will answer this question in two steps: In Section 4, after having provided more details concerning the technical side of the AI pseudo-annotator, we will discuss its performance from a Natural Language Processing (NLP) perspective; in Section 5, we will present the results of a computer-assisted annotation experiment in which the AI will be employed to suggest relevant claims to the annotators (and not just to the experts in the gold merging stage).

4. The AI-Pseudo-Annotator: NLP Support for Claim Identification and Categorisation

This section describes the AI pseudo-annotator. It is responsible for the tasks of claim detection and claim mapping (categorisation), both implemented as (supervised) classification. Classification is the task of assigning an input to a set of pre-defined categories. We approach claim identification as a token sequence labelling task with a variant of the BIO schema (Ramshaw & Marcus, 1999). Specifically, the input to the identifier model consists of a sentence, represented as a word sequence (for practical reasons, sentence length is limited to 128 words). The claim identifier labels each word in the input with a tag from the list of B(eginning of)-CLAIM, I(nside)-CLAIM, O(utside) the claim. Claim classification is realised as a multi-label classification for each word sequence that was predicted to be a claim: The classifier assigns one or more theoretically motivated classes—as defined in the codebook—to the sequence. Note that we currently do not automatically recognise actors. To extract claim-author pairs we, therefore, adopt an 'oracle' setting where we pair up all claims that were correctly recognised automatically with their corresponding manually annotated actors.

In what follows, we provide a brief description of the dataset, the annotation scheme, data representation, and the machine learning methods we apply for the AI annotator. The description is aimed primarily at NLP experts to enable replication of our approach (see Alpaydin, 2009, for an accessible introduction to machine learning in general).

1) Dataset and classification scheme: Our dataset consists of all articles published in 2015 in the German newspaper *taz—die tageszeitung* on the issue of migration in Germany (about 2000 articles). It is steadily expanded and contains so far over 1000 fully annotated articles with more than 4500 claims (an earlier version is already freely available). We have designated a fixed, randomly drawn set of 15% of the articles as a test set. The remaining 85% of the articles serve as the training set. It contains 342 articles consisting of 12,571 sentences and 1400-word sequences are labelled as a claim. The average claim length in the training set is 20.12 words per claim. Similarly, our test set contains 159 articles, 1753 sentences and 159 claims where the mean claim length is 19.13.

The annotation schema contains 8 higher-level categories (controlling migration; residency; integration; domestic security; foreign policy; economy; society; and procedures) as well as finer-grained categories (e.g., accommodation as an integration strategy). We currently only perform automatic classification on the higher-level categories. It is not possible to classify all fine-grained categories at the desired quality. This is not a fundamental problem of granularity. Rather, it is a practical problem of (not) having a sufficient number of examples for each fine-grained class to learn reliable classifiers for them. Even the distribution of the higher-level categories is fairly skewed, as is usual in language data. We would expect more annotated examples to improve classification quality. However, idiosyncrasies of the categories also need to be taken into account. Categories with a specific technical jargon (e.g., Dublin Procedure) are generally easy to learn from a few examples, while other categories may require more examples (e.g., limiting migration). Generally speaking, what we see here is a trade-off between the interest of political science in developing detailed and specific analyses of individual debates and the annotation effort that is necessary to annotate corpora with the resulting detailed codebooks.

2) Representation and classification: The MARDY system builds on the state-of-the-art approach in NLP to model semantics that uses low-dimensional, dense vectors—so-called embeddings—to represent words (and other linguistic entities). Embeddings can be learned automatically from large corpora by exploiting the distributional hypothesis, which states that words that occur in similar contexts have similar meanings (Firth, 1957). Currently, the best performance is generally achieved with contextualised embeddings (Devlin, Chang, Lee, & Toutanova, 2019; Peters et al., 2018) obtained with deep neural models, mostly based on an architecture called Transformer (Vaswani et al., 2017) and trained on huge amounts of raw texts. There are many publicly available pre-trained models that can be used for obtaining contextualised word embeddings.

4.1. Developing Claim Identification and Classification Methods

Our claim classifier is an update of the BERT model presented in Padó et al. (2019). Similar to the earlier model, it is based on the BERT Transformer (Devlin et al., 2019). However, we made the model more language specific which leads to a modest increase in quality (see below for details). Specifically, we use the Deepset German BERT model (Deepset GmbH, 2019), which was trained on large German corpora, including Wikipedia. Next, we fine-tune the contextualised embeddings on our complete *taz* newspaper corpus (all *taz* articles in 2005, 2010, and 2015), consisting of 3,258,697 sentences and 58,411,202 words, using next sentence prediction loss as the pretraining objective. Finally, we train the claim identifier using the 342 articles in our training set.

We use individual sentences as input to the claim identifier and process the input as suggested in the original BERT paper (Devlin et al., 2019); the input text is split into word pieces before being fed to the BERT model. The resulting token sequence that is used for classification is typically longer than the word sequence of the sentence. We ignore the predictions made for subunits during loss calculation in training and in evaluation. The classes (B-Claim, I-Claim, and O, as defined above) are assigned with a standard softmax layer. We use the Adam optimiser with learning rates of 2e-5, $\beta 1 = 0.9$, $\beta 2 = 0.999$, a batch size of 16 and a dropout with p = 0.5 on all layers. We train the classifier for seven epochs and store models and evaluation results after each iteration. As the final model, we select the model with the highest development set recall value among the subset of saved models where the recall/precision ratio is equal to, or smaller than, two. This procedure leads the claim identifier to over generate claims to some extent—a trade-off that we believe is sensible in our current pipeline architecture.

For claim classification, we assume that claims have already been identified. Each claim is assigned one or more of the eight top-level categories of the MARDY claim codebook. The basic architecture of the claim classification model is very similar to the claim identifier: again, we use a fine-tuned version of BERT to obtain contextualised embeddings. We use the Adam optimiser with learning rates of 5e-5, $\beta 1 = 0.9$, $\beta 2 = 0.999$, a batch size of 32 and a dropout with p = 0.1 on all layers. We train the classifier for seven epochs and select the model with the best macro-averaged F1 score on the development set (i.e., the model is optimised to find a good trade-off between precision and recall). The main difference is that claim classification is an instance of multi-label classification (i.e., more than one claim class can be assigned to each claim). We handle this change by replacing the softmax layer with a sigmoid layer, as a result of which multiple classes can be assigned at the same time.

4.2. Evaluation of Classifier Quality

Evaluation of classification tasks is typically carried out by computing per-class precision, recall, and F1 scores (Jurafsky & Martin, 2009). For each class T, precision measures what percentage of predictions of T is correct, while recall measures what percentage of instances of T is recovered. F1 score is the harmonic mean of precision and recall. For claim identification, we report token level precision, recall, and F1 score for the claim class. We evaluate claim classification on the test set by comparing predictions to gold standard claims and report results macro-averaged across the eight major claim categories in the dataset at the claim level. This use of a single held-out test set is standard practice in computational linguistics; an alternative would have been to use n-fold cross validation.

Table 1 lists the results of evaluating the claim identifier and classifier on the test portion of our annotated dataset. The results show that the model delivers reasonable predictions, in particular at the claim identification level. Given that we select the claim identification model to maximise recall, it is not surprising that precision is somewhat lower, but it is still at a useful level. For the claim classifier, where we instead select the model with the best overall score, precision and recall are considerably more balanced. Given that claim classification is a multi-label classification task, we consider this a promising result. To establish a comparison to previous work, the last two rows of Table 1 present results for the best claim identification (EmbTAZ:w,c+BiLSTM+CRF) and claim classification (BERT) models from Padó et al. (2019) when evaluated on our current dataset. Our current claim identification model performs two points F-score higher, with increases both in terms of precision and recall due to the better language specific pre-training. Similarly, our claim classifier performs better in terms of all metrics, with particular increases in macro averaged F-score and Recall. An additional advantage is that both classifiers now use the same overall architecture.

We believe that a high recall and a lower precision form a reasonable trade-off for semi-automatic annotation support, since human coders review the machine predictions and can therefore correct precision errors, while due to the high recall the model has a chance of finding instances which may be missed by human annotators. Note that evaluation results are always relative to the similarity of the training and test data: Since these are both drawn from the *taz* corpus and from documents with the same topic, we would expect similar results for other *taz* articles, but possibly lower results when the classifiers are applied to other corpora or other topics. This is not a problem of our specific approach, but a problem that applies in general to NLP and supervised ma-

Table 1. Precision, recall, and F1 scores of automatic models.

	Precision	Recall	F1 score
Claim Identification	0.39	0.77	0.52
Claim Classification	0.65	0.56	0.60
Claim Identification (Padó et al., 2019)	0.37	0.73	0.50
Claim Classification (Padó et al., 2019)	0.61	0.46	0.52

Note: Claim identification (at token level, for class ‘claim’); claim classification (at claim level, macro averaged across classes).

chine learning: models lose quality with increasing distance between the data they were trained on and the data they are applied to.

Another potential concern is whether the automatic models are fair in the sense of not exhibiting better quality for some parts of the data than for other parts (Binns, 2018); this topic has received substantial attention in NLP in previous years (Hovy & Spruit, 2016). Since the list of such covariates of quality is open-ended, we cannot rule out a problem of this type in principle. However, we carried out two analyses. First, we checked for the influence of the political affiliation of the actor on the recall of claim identification. We did so by computing a contingency table with the true positives and false negatives of our model for each set of actors affiliated to a political party (FDP [Free Democratic Party], CDU [Christian Democratic Union], CSU [Christian Social Union], SPD [Social Democratic Party], Green Party, Left Party, and AfD [Alternative for Germany]), plus the set of unaffiliated actors, as defined by Wikidata. We carried out Fisher's Exact Test on this contingency table and did not find an influence of affiliation on recall ($n = 251$, $p = 0.83$). Second, we investigated whether the claim identifier was able to generalise properly to novel claims not encountered in the training set. To do so, we defined a claim as 'seen' if the combination of actor and category occurred in the training set (this holds for 13.4% of the claims in our test set). We found that the recall of claim identification was 94.8% on seen claims and 74.2% on unseen claims. We conclude that the model performs somewhat better on previously seen claims. However, the quality of novel claims is still decent enough to indicate that the model is able to generalise to unseen data. Therefore, its overall quality cannot be explained only by memorisation of the training data. With this in mind and based on the improved results compared to earlier models (cf. Table 1), the next section tackles the question of how these improvements and the general approach translate into the annotation process in practice.

5. Annotation Experiment

We conducted an experiment in order to test whether the support by the AI pseudo-annotator leads to an increase in annotator performance, i.e., whether it speeds up annotation (see Section 3, Goal 1) and whether it increases annotation quality (see Section 3, Goal 3). The experiment follows a design in which two separate groups are repeatedly exposed to either treatment or to no treatment over four rounds in an alternating manner (Table 2). Annotation speed is measured in average annotation time per claim, quality by computing recall, and precision and F1 scores. The articles used for this experiment are disjoint from the complete 'gold standard' dataset (comprising of the training and test sections) as described in Section 4 above. This is obviously necessary in order to avoid that annotators may remember articles that they annotated previously. Since articles for the experiment were also drawn randomly from the corpus, similar to the test set, we believe that the classifier accuracy and fairness results presented in Section 4 carry over to this dataset as well.

The participants were six experienced annotators (two senior researchers and four student assistants), who were familiar with the annotation environment and trained on the topic. The participants were assigned to group A or B (group sizes $n = 3$). We balanced the groups with respect to the number of training hours and to prevent the senior researchers to be in the same group. Depending on the group, the participants were exposed to the treatment, consisting of AI suggestions based on predictions from the classifier, or no treatment. In both cases, annotators were asked to read and manually annotate the articles. The only difference was that the treatment group was able to immediately use the pre-annotated claims from the AI pseudo-annotator. The experiment took place on the campus of the University of Bremen and ran over the course of two days and four rounds. In the first round, Group A started annotating with suggestions by the AI and Group B without (Table 2). This setting was reversed in round 2. To account for fatigue (Ellis, 1999, p. 556), the order of exposure/non-exposure per group was switched on day two (Rounds 3 and 4, respectively). This setting allows us not only to compare differences between groups but also within subjects (Ellis, 1999). In each round, ten articles had to be annotated with a time limit of 105 minutes per round, a reasonable choice given previous knowledge about typical annotation durations. The articles in each set were similar with respect to length, difficulty, and claim frequency, facilitating between-group comparisons.

In this experiment, we asked annotators to only identify and classify the claim, in order to isolate the effect of the AI support on claim detection. Information about actors and polarity was added later and thus is not part of the experiment.

The small number of participants and the involvement of the researchers limits the generalisability of the

Table 2. Design of the experiment.

	Group A	Group B
Day 1, Round 1	Treatment	No Treatment
Day 1, Round 2	No Treatment	Treatment
Day 2, Round 3	No Treatment	Treatment
Day 2, Round 4	Treatment	No Treatment

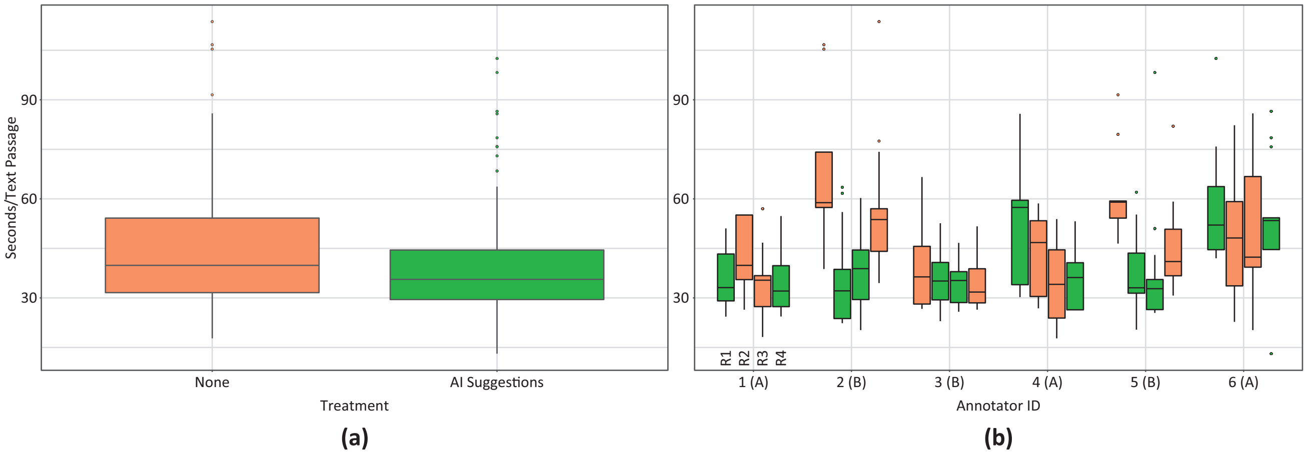

Figure 3. Annotation time per text passage. Note: Figure 3a: median time treatment vs. no treatment; Figure 3b: median time per annotator and round.

results, but it still offers a first impression of the effects that the introduction of an AI pseudo-annotator can have on human annotators. Firstly, Figure 3 shows that the support of machine learning during annotation leads to a marginal increase in annotation speed. Secondly, Table 3 demonstrates that the treatment group with suggestions from the pseudo-annotator shows much higher recall scores and an insignificant decrease in precision. Lastly, we observe a moderate increase of inter-annotator-agreement. Overall, the pseudo-annotator offers promising yet not always accurate suggestions. Over the course of the experiment, the participants annotated a total of 2555 text passages (425.8 on average) containing 3114 claims (519 on average). This resulted in a gold-standard encompassing 573 claims spread over 453 text snippets. The pseudo-annotator made 467 suggestions. Of these, 331 were accepted into the gold-standard (70.9%).

Overall, the experiment shows a slight decrease in the median value of annotation time per text passage, but the difference is not very pronounced, dropping about 10% from 39.9 to 35.6 seconds (Figure 3a). Looking at the measures for individual annotators (Figure 3b), we see that the overall gain is mainly the result of significant speed gains for two of the six annotators (ID 2, student, and ID 5, senior), while the AI support made hardly any difference for annotators 1 and 3 (both students) and for annotators 4 (senior) and 6 (student) the average time to annotate a claim with AI support was even slightly higher than without support. This shows a substantial amount of personal variation regarding the use of automatically generated suggestions.

A more rigorous statistical analysis on the basis of a fixed-effects-regression confirms these results (see Appendix 2 in the Supplementary File). More specifically, we controlled for unobserved factors (e.g., intelligence), which might fluctuate across annotators by introducing fixed effects for each participant and additionally a time trend for rounds to account for learning effects. Moreover, we included the number of claims found by each annotator per article and the article length (in tokens). The regression analysis confirms that the speed gain from pseudo-annotator suggestions is not statistically significant, and the effect size itself is rather small. During the experiment, each participant saves on average about 42 seconds per article when having access to predictions compared to the case of manual annotation (see model 4 in Appendix 2 in the Supplementary File). Annotation with AI support is thus on average about 10% quicker than without.

To assess the impact on annotation quality, Table 3 looks at recall, precision, and F1 score. We see the following results: Average precision with AI support is minimally lower than without support (0.81 vs. 0.82) but recall increases substantially and gains over five points (0.74 to 0.80). In fact, all annotators without exception exhibit a higher recall with the support of the pseudo-annotator. Together, both changes lead to an overall increase in the F1 score from 0.77 to 0.80. Out of the six annotators, four were able to increase their overall an-

Table 3. Precision, recall, and F1 score in the experiment.

Annotator (Group)	ø		1 (A)		2 (B)		3 (B)		4 (A)		5 (B)		6 (A)		*AI*
Treatment	none	AI	none	AI	none	AI	none	AI	none	AI	none	AI	none	AI	—
Recall	0.74	0.80	0.84	0.90	0.65	0.76	0.77	0.80	0.82	0.86	0.60	0.69	0.77	0.80	*0.73*
Precision	0.82	0.81	0.83	0.83	0.89	0.88	0.90	0.84	0.72	0.65	0.90	0.88	0.68	0.81	*0.71*
F1 score	0.77	0.80	0.84	0.86	0.75	0.81	0.83	0.82	0.77	0.74	0.71	0.77	0.72	0.80	*0.72*

notation quality in terms of F1 score. Annotation quality of the remaining two annotators (3 and 4) deteriorates slightly to moderately.

Overall, the results of the experiment suggest that the integration of machine learning suggestions into the annotation workflow improves annotation (at least recall and F1 score), but the speed gain is only relatively small, especially if we account for the additional time that is needed to train the AI. On its own (last column of Table 3), the AI pseudo-annotator is reasonably good but still less accurate than the average human annotator, and thus cannot replace them yet—at least if we are interested in correctly identifying all claims in a given set of texts. The remaining question, however, is if the AI annotator is good enough to build reliable discourse network representations—this is exactly the goal of the modelling experiment we report in the following section.

6. Discourse Networks

Annotating all relevant claims in newspaper articles produces data with a certain amount of redundancy because both political actors and journalists tend to repeat themselves: If an article reports three times about claim X from actor A, two times about claim Y from actor B and only once about claim Z from actor C, it effectively reports information on three different actor-claim dyads. In the evaluation approach which characterised the previous sections, a claim annotation tool would have to identify all six occurrences of these claims to get full credit, while such (near)-repetitions are often ignored in DNA because they do not provide substantial new information: Only one instance of each actor-claim dyad has to be detected. This indicates that network construction can proceed even based on a (somewhat) incomplete annotation. Often DNA studies even normalise edge-weights of actor-claim dyads across multiple articles per day, so that one specific claim from one specific actor is counted only once per day. All additional mentions of the same actor-claim dyad in the same or in other articles on this day are treated as redundant.

Figure 4 represents the network of all actors and claims present in the gold standard annotation, created from the manual and AI annotations of the 40 articles of the experiment data set. Claims not found by the AI pseudo-annotator, i.e., the AI's false negatives, and the actors that appear only in those claims are highlighted in red. In line with the expectation that the network may be less sensitive to false negatives, we find that the AI detects 77.2% of all edges, which is a four points higher

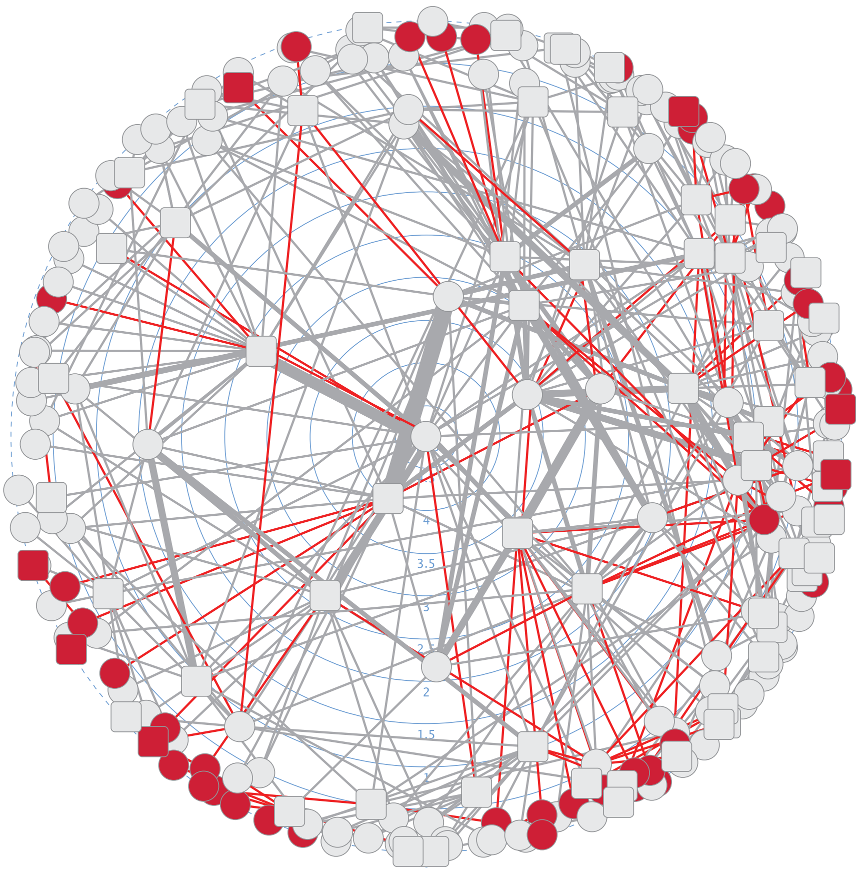

Figure 4. Discourse network from the experiment data, containing all actor-claim dyads of the manually created gold standard. Notes: Circles represent actors, squares represent claims. Actors, claims, and edges not found by the AI are highlighted in red. Placement of the nodes represents their eigenvector centrality value, nodes with higher centrality values are placed more centrally.

recall value compared to overall claim detection. More importantly, only seven out of the 77 claim categories are never detected by the AI and 80.1% of the actors are present in the AI's true positive claims (but as mentioned above, actors were manually annotated in the gold standard only, so they were not really found by the AI). Nodes not found in the AI's true positive claims account for less than 6.4% of the network's overall eigenvector centrality. In other words, the nodes not present in the AI set are mostly only marginal nodes in the network. Figure 4 illustrates this by placing nodes with higher eigenvector centrality values in the centre of the graph and nodes with low centrality values at the margins.

To further evaluate the accuracy of the AI suggestions, we can restrict our analysis to the network core, instead of looking at the complete network. There are several options to determine network cores. We use a very simple method that is particularly useful for bipartite weighted networks. In our network, we assign edge values to the actor-claim dyads that correspond to the number of occurrences of this dyad on separate dates in our data. So, if actor A makes claim X on day 1, 2 and 3, the A–X actor-claim dyad gets the value of 3. We now create a core network that consists of all edges and adjacent nodes where edge values are greater than one—a two-slice of our original network (de Nooy, Mrvar, & Batagelj, 2005, p. 98). On a substantial level, this core network contains all actors and claims for which the same actor-claim dyad was reported at least twice for different dates. It is reasonable to assume that normally only actors whose claims are reported more than once in a certain time period can have an influence on the future direction and the outcome of a political debate.

The result can be seen in Figure 5. This two-slice of the entire network captures and displays the core of the underlying discourse structure as reported in the 40 randomly selected articles of the experiment. On a substantial level, the actors and claims in the core network are no surprise for an avid observer of the 2015 migration debate in Germany. They comprise government and opposition parties and prominent political actors addressing issues that dominated the discourse in this year. But since our data set only contains 40 randomly selected articles, our focus here is not on the substantial validity of the observed discourse network.

In the context of our experiment, the much more interesting result is that the AI pseudo-annotator has found all claims in the core network. At the two-slice level, the AI is able to completely reproduce the network based on the manually annotated gold standard. Recall at this level is 100%, if we ask the system to only detect and not yet classify the claims and if we discount for the fact that automatic speaker identification is not yet implemented in the current prototype system. So far, this does not mean that we can generate core discourse networks in a fully automated process, since fine-grained categories, actor names and polarity of the claims have been added manually in the experimental setting. But the result suggests that the AI suggestions could be used in a much more far-reaching computer supported anno-

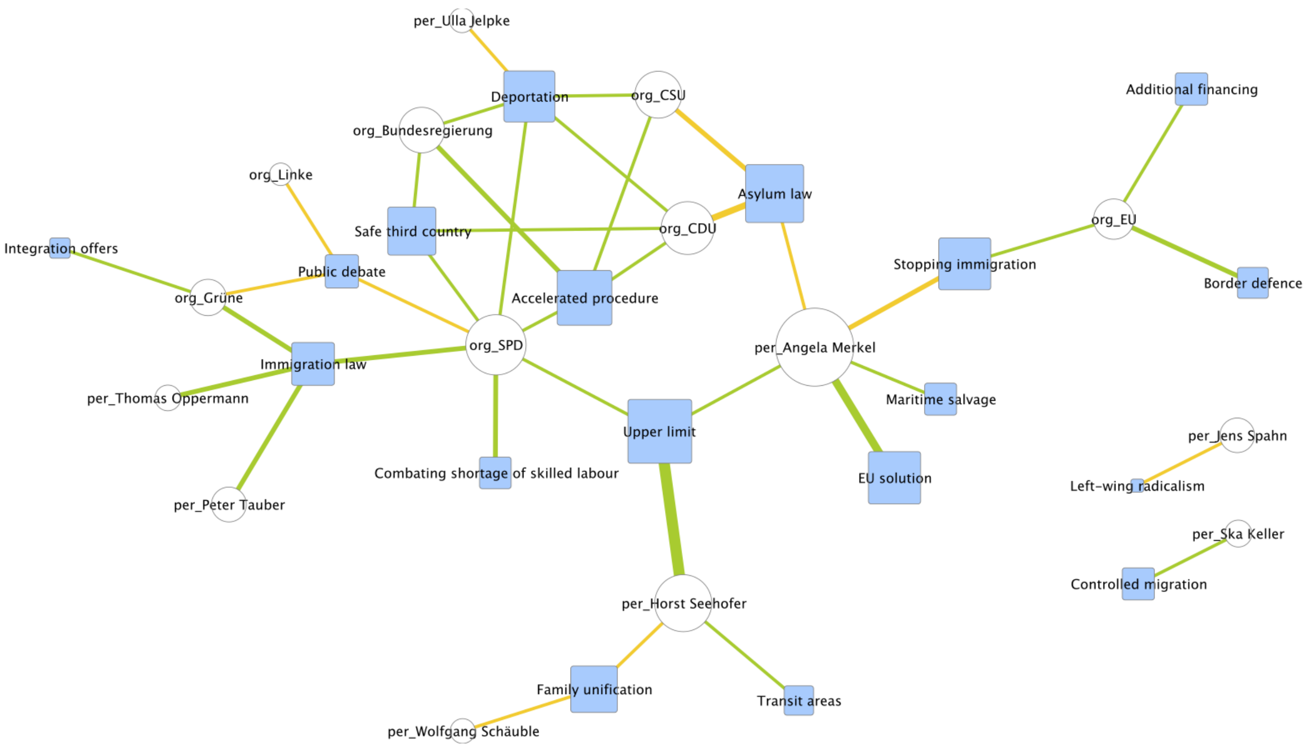

Figure 5. Two-slice of the discourse network from Figure 4 containing all actor-claim dyads mentioned at least twice in our data set. Notes: Circles represent actors, squares represent claims. Green edges represent support for the claim, orange edges opposition to the claim.

tation scenario: Instead of offering the human annotators AI suggestions, but still asking them to read the complete text of each article, a setting becomes feasible in which the human annotator has only to decide which of the AI suggestions should be accepted and which should be rejected. Since there are no false negatives in AI suggestions at the two-slice level, human annotators would only have to weed out the false positive AI suggestions in order to get an accurate representation of the core discourse network. This task is identical to creating the gold standard from the manual annotations. Limiting the manual annotation work to only this remaining task would drastically reduce the time spent on the typically laborious annotation process. Of course, this would still require the manual annotation of a large enough training set for the AI.

7. Conclusion

While the integration of machine learning in annotation workflows has been suggested before, no working systems have yet been developed that leverage machine learning not only for corpus creation and text selection but also for the actual annotation of texts using complex and multifaceted abstract categories. The MARDY annotation environment described in this article strives to offer such an integrated system.

In order to evaluate how useful such a system can be for an extensive annotation task in a research project focusing on current political debates, we have tested the performance of the system in an annotation experiment. The results show that a system can be trained to provide machine learning based annotation suggestions which improve the performance of human annotators, both in annotation speed and regarding the F1 score of annotation quality. Adding an AI pseudo-annotator thus can help to ease the time and labour-intensive task of manual annotation. However, the gains on this level are limited and it is questionable whether the additional time and expertise needed to provide AI suggestions at a sufficient level of accuracy outweigh the time and resources gained in the annotation process.

But a closer examination of the data produced by the AI pseudo-annotator reveals surprising and promising results on another level. Our results show that if we account for the fact that newspaper articles contain a significant amount of redundant information about political claims-making, and if we use the structural perspective of the discourse network approach to identify central actors and claims of a political debate, we can use an AI pseudo-annotator to provide information about the core discourse network with a very high level of recall and without compromising precision. This opens the possibility of an annotation system in which human annotators no longer have to read the complete text but only have to weed out the false positive AI suggestions.

In future experiments, the robustness of our findings has to be assessed. Open questions are to what extent an AI trained on texts from one newspaper is also able to predict claims in other news sources, whether claim prediction quality and the system's ability to recover core discourse networks differs across issues or depends on issue salience (and thus the volume of articles on an issue per time period) or the level of contention. Also, the observed differences between the annotators with support from the AI pseudo-annotator merit further investigation. MARDY is still a prototype and not a ready-to-use tool, but the description of its elements and the annotated data published with this article hopefully will help the scientific community to move forward in creating tools that allow for more detailed analyses of large text corpora in the social sciences.

A demo version of the MARDY system can be accessed at http://hdl.handle.net/11022/1007-0000-0007-DF36-2. This page also offers tutorial videos and a more detailed manual for the annotation environment, links to the documentation of the classifier code, to the classifier demo, the R scripts for experiment, and network analysis and to the data.

Acknowledgments

The research presented here was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within the SPP RATIO—Project No. 375875969.

Conflict of Interests

The authors declare no conflict of interests.

Supplementary Material

Supplementary material for this article is available online in the format provided by the authors (unedited).

References

Alpaydin, E. (2009). *Introduction to machine learning*. Cambridge, MA: MIT Press.

Binns, R. (2018). Fairness in machine learning: Lessons from political philosophy. In S. A. Friedler & C. Wilson (Eds.), *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, *PLMR 81* (pp. 149–159). New York, NY: Proceedings of Machine Learning Research.

Blessing, A., Blokker, N., Haunss, S., Kuhn, J., Lapesa, G., & Padó, S. (2019). An environment for relational annotation of political debates. In M. R. Costa-Jussà & E. Alfonseca (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System demonstrations* (pp. 105–110). Florence: Association for Computational Linguistics.

Burscher, B., Vliegenthart, R., & de Vreese, C. H. (2016). Frames beyond words: Applying cluster and sentiment analysis to news coverage of the nuclear power issue. *Social Science Computer Review*, *34*(5), 530–545.

D'Angelo, P., & Kuypers, J. A. (Eds.). (2010). *Doing news framing analysis: Empirical and theoretical perspectives*. New York, NY: Routledge.

de Nooy, W., Mrvar, A., & Batagelj, V. (2005). *Exploratory social network analysis with Pajek*. Cambridge: Cambridge University Press.

Deepset GmbH. (2019). Open sourcing German BERT. *Deepset*. Retrieved from https://deepset.ai/german-bert

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & Thamar Solorio (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies* (pp. 4171–4186). Minneapolis, MN: Association for Computational Linguistics.

Ellis, M. V. (1999). Repeated measures designs. *The Counseling Psychologist*, *27*(4), 552–578.

Firth, J. R. (1957). A synopsis of linguistic theory, 1930–1955. In J. R. Firth (Ed.), *Studies in linguistic analysis* (pp. 1–32). Oxford: Blackwell.

Haunss, S. (2017). (De-)legitimating discourse networks: Smoke without fire? In S. Schneider, H. Schmidtke, S. Haunss, & J. Gronau (Eds.), *Capitalism and its legitimacy in times of crisis* (pp. 191–220). Basingstoke: Palgrave Macmillan.

Hovy, D., & Spruit, S. (2016). The social impact of natural language processing. In K. Erk & N. A. Smith (Eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (pp. 591–598). Berlin: Association for Computational Linguistics.

Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing* (2nd ed.) Upper Saddle River, NJ: Prentice-Hall.

Kelle, U. (2008). *Die Integration qualitativer und quantitativer Methoden in der empirischen Sozialforschung: Theoretische Grundlagen und methodologische Konzepte* [The integration of qualitative and quantitative methods in empirical social research: Theoretical foundations and methodological concepts] (2nd ed.). Wiesbaden: VS Verlag.

Klüver, H. (2009). Measuring interest group influence using quantitative text analysis. *European Union Politics*, *10*(4), 535–549.

Koopmans, R., & Statham, P. (Eds.). (2010). *The making of a European public sphere: Media discourse and political contention*. Cambridge: Cambridge University Press.

Kuckartz, U. (2014). *Mixed methods: Methodologie, Forschungsdesigns und Analyseverfahren* [Mixed methods: Methodology, research designs and analysis]. Wiesbaden: VS Verlag.

Laver, M., Benoit, K., & Garry, J. (2003). Extracting policy positions from political texts using words as data. *American Political Science Review*, *97*(2), 311–331.

Leifeld, P. (2009). Die Untersuchung von Diskursnetzwerken mit dem Discourse Network Analyzer (DNA) [The analysis of discourse networks with the discourse network analyzer (DNA)]. In V. Schneider, F. Janning, P. Leifeld, & T. Malang (Eds.), *Politiknetzwerke: Modelle, Anwendungen und Visualisierungen* [Policy networks: Models, applications and visualizations] (pp. 391–404). Wiesbaden: VS Verlag.

Leifeld, P. (2016). *Policy debates as dynamic networks: German pension politics and privatization discourse*. Frankfurt: Campus.

Nagel, M., & Satoh, K. (2019). Protesting iconic megaprojects: A discourse network analysis of the evolution of the conflict over Stuttgart 21. *Urban Studies*, *56*(8), 1681–1700.

Padó, S., Blessing, A., Blokker, N., Dayanik, E., Haunss, S., & Kuhn, J. (2019). Who sides with whom? Towards computational construction of discourse networks for political debates. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the annual meeting of the association for computational linguistics* (pp. 2841–2847). Florence: Association for Computational Linguistics.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In M. Walker, H. Ji, & A. Stent (Eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies* (pp. 2227–2237). New Orleans, LA: Association for Computational Linguistics.

Ramshaw, L. A., & Marcus, M. P. (1999). Text chunking using transformation-based learning. In S. Armstrong, K. Church, P. Isabelle, S. Manzi, & E. Tzoukermann (Eds.), *Natural language processing using very large corpora* (pp. 157–176). Dordrecht: Springer.

Rädiker, S., & Kuckartz, U. (2019). *Analyse qualitativer Daten mit MAXQDA: Text, Audio und Video* [Analysis of qualitative data with MAXQDA: Text, audio and video]. Wiesbaden: VS Verlag.

Schmidtke, H., & Nullmeier, F. (2011). Political valuation analysis and the legitimacy of international organizations. *German Policy Studies*, *7*(3), 117–153.

Stulpe, A., & Lemke, M. (2016). Blended reading. In M. Lemke & G. Wiedemann (Eds.), *Text Mining in den Sozialwissenschaften: Grundlagen und Anwendungen zwischen qualitativer und quantitativer Diskursanalyse* [Text mining in the social sciences: Fundamentals and Applications between qualitative and quantitative discourse analysis] (pp. 17–61). Wiesbaden: Springer Fachmedien Wiesbaden.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Proceedings of advances in neural information processing systems* 30 (pp. 5998–6008). Long Beach, CA: Neural Information Processing Systems Foundation.

Vrandečić, D., & Krötzsch, M. (2014). Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, *57*(10), 78–85.

Walter, D., & Ophir, Y. (2019). News frame analysis: An inductive mixed-method computational approach. *Communication Methods and Measures*, *13*(4), 248–266.

Wang, C., & Wang, L. (2017). Unfolding policies for innovation intermediaries in China: A discourse network analysis. *Science and Public Policy*, *44*(3), 354–368.

Welbers, K., van Atteveldt, W., & Benoit, K. (2017). Text analysis in R. *Communication Methods and Measures*, *11*(4), 245–265.

Wiedemann, G. (2016). *Text mining for qualitative data analysis in the social sciences*. Wiesbaden: Springer.

Wilkerson, J., & Casas, A. (2017). Large-scale computerized text analysis in political science: Opportunities and challenges. *Annual Review of Political Science*, *20*(1), 529–544.

About the Authors

Sebastian Haunss is Professor in Political Science and Head of the research group on Social Conflicts at the Research Center on Inequality and Social Policy at the University of Bremen (SOCIUM). He has published on social movements, discourse dynamics, social networks, mixed methods, political legitimation, and conflicts of the knowledge society.

Jonas Kuhn is Professor of Computational Linguistics at the University of Stuttgart. After completing his Doctorate in Stuttgart in 2001, he was a Postdoc at Stanford University, Assistant Professor at the University of Texas at Austin, led a junior research group at Saarland University, and took up a full professorship at the University of Potsdam in 2006. He holds his current position since 2010. Kuhn's research interests range broadly from linguistically informed data-driven models in Natural Language Processing to the development of cross-disciplinary methods for text analysis in the humanities and social sciences.

Sebastian Padó is a Professor of Computational Linguistics at Stuttgart University. He studied in Saarbrücken and Edinburgh, receiving his MSc in 2002 and PhD in 2007. After a time as a Postdoctoral Scholar at Stanford, he has held Professor positions in Heidelberg (2010–2013) and Stuttgart (since 2013). His core research concerns learning, representing, and processing semantic knowledge from and in text. Examples include distributional models of linguistic concepts, multilingual modelling, discourse structure, and applications of semantics in computational humanities and social sciences.

Andre Blessing is a Postdoctoral Researcher in Computational Linguistics at the Institute for Natural Language Processing at the University of Stuttgart. His research interests include digital humanities, information extraction, and text mining.

Nico Blokker is a PhD Candidate at the Research Center on Inequality and Social Policy at the University of Bremen. His main research interests are computational social science, social network analysis, and text analysis. In his dissertation, he focuses on (latent) spatial representations of discourse dynamics.

Erenay Dayanik is a PhD Student at the University of Stuttgart, in the Institute for Natural Language Processing. He obtained his MSc in Computer Engineering from the Artificial Intelligence Lab, Koc University, Istanbul in 2018, and his BSc in Computer Engineering from METU, Ankara, in 2015. His research interests focus on deep learning and natural language understanding. He has published papers at conferences and in journals on NLP, including the Association for Computational Linguistics and the Conference on Computational Natural Language Learning.

Gabriella Lapesa is a Postdoctoral Researcher at the Institute for Natural Language Processing at the University of Stuttgart. She holds a BA in Digital Humanities and a MA in Theoretical and Applied Linguistics, both from the University of Pisa, and a PhD in Cognitive Science from the University of Osnabrück. She is interested in the development of interdisciplinary NLP methods at the interface with computational social science, theoretical linguistics, and cognitive science.